Department of Computer Engineering

Academic Term: First Term 2023-24

Class: T.E /Computer Sem – V / Software Engineering

Practical No:	**1**
Title:	**Software Requirement Specification**
Date of Performance:	**26 / 07 / 2023**
Roll No:	**9542**
Team Members:	**Group 8**

Rubrics for Evaluation:

Sr. No	**Performance Indicator**	**Excellent**	**Good**	**Below Average**	**Total Score**
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

Signature of the Teacher:

Department of Computer Engineering

Academic Term: First Term 2022-23

Class: T.E /Computer Sem – V / Software Engineering

Signature of the Teacher:

Case Study 1 - Hospido: Revolutionising Healthcare Accessibility and Connectivity

1.Abstract

The coronavirus pandemic brought the world to its knees, hitting the medical community hardest. Doctors worked over 2x their usual hours, making extraordinary contributions to meet urgent patient needs. In response, Hospido, a healthcare app, emerged to provide round-the-clock medical services and connect patients with specialised doctors. The healthcare industry is witnessing a transformative phase, with unprecedented collective efforts focused on building innovative healthcare products. Hospido's single-click hospital alert system and online appointment booking services have become crucial tools in this challenging time.

2.Introduction

2.1 Purpose

The purpose of this project is to revolutionise healthcare accessibility by providing a user-friendly application, Hospido, that offers round-the-clock medical services and connects patients with specialised doctors instantly.

2.2 Scope

The scope of this project includes developing and implementing the Hospido application, which will facilitate access to healthcare services 24/7, allowing patients to connect with specialised doctors and receive immediate medical attention. The app will also provide essential information about hospitals, their services, available beds, and doctors on duty, enhancing overall healthcare accessibility and efficiency.

2.3 Developer’s Responsibilities

Developer's Responsibilities for Hospido:

1. Designing and Developing the Application: The developer is responsible for creating a user-friendly and intuitive interface for the Hospido application, ensuring seamless navigation and functionality.

2. Implementing 24/7 Medical Services: The developer will integrate features that allow patients to access healthcare services at any time, enabling them to request immediate medical attention and alert top hospitals about their arrival.

3. Connecting Patients with Specialised Doctors: The developer will build a robust system that matches patients with the most appropriate specialists based on their medical needs and provides online appointment booking services.

4. Providing Hospital Information: Ensuring that the app displays accurate and up-to-date information about various hospitals, including the services offered, available beds, and the doctors currently on duty.

5. Ensuring Data Security and Privacy: Implementing strong data security measures to protect patient information and ensuring compliance with healthcare privacy regulations.

6. Conducting Testing and Bug Fixing: Thoroughly testing the application to identify and fix any bugs or issues, ensuring a smooth user experience.

7. Continuous Maintenance and Updates: The developer will be responsible for regular maintenance and updates to keep the application running efficiently and incorporating user feedback for improvement.

3.General Description

3.1 Product Functions Overview

Hospido is a groundbreaking healthcare application that aims to revolutionise the way patients access medical services. With a user-friendly interface, it offers 24/7 medical assistance, allowing patients to receive immediate attention with just a click. The app efficiently connects patients with specialised doctors, ensuring personalised care and timely appointments for their unique medical needs. It also provides vital information about various hospitals, such as services offered, available beds, and doctors on duty. During emergencies, Hospido's hospital alert system enables patients to notify top hospitals in their area about their immediate arrival, streamlining the admission process. Moreover, the application offers seamless online appointment booking services, empowering patients to schedule appointments with their preferred specialists at their convenience. Data security and privacy are paramount, with the implementation of robust measures to safeguard patient information and comply with healthcare regulations. Regular updates and maintenance guarantee a smooth experience, making Hospido a reliable and accessible platform for medical care anytime, anywhere.

3.2 User Characteristics

Hospido caters to a diverse range of users with specific healthcare needs. The primary users are patients seeking medical services, including those with urgent health requirements or requiring specialised care. Healthcare professionals, such as doctors and nurses, utilise the platform to attend to patients and manage their appointments efficiently. Hospital administrators and staff also play a crucial role, using Hospido to update and maintain hospital information, ensuring accurate details are available to patients. During emergencies, the app becomes vital for users who need to quickly notify top hospitals and receive prompt medical attention. Hospido is designed to appeal to tech-savvy individuals comfortable with mobile applications and technology for accessing healthcare services. Additionally, patients looking for efficient and convenient healthcare solutions are attracted to Hospido's online appointment booking feature. The app's commitment to data security and privacy also reassures privacy-conscious individuals. By understanding these user characteristics, Hospido aims to provide a seamless and satisfactory healthcare experience to all its users.

4 Specific Requirements

4.1 Software Constraints

It sounds like you're describing a healthcare app called Hospido that was developed in response to the challenges posed by the coronavirus pandemic. The app aims to provide round-the-clock medical services and connect patients with specialised doctors. It offers features such as a single-click hospital alert system and online appointment booking services. This app seems to be playing a significant role in the current healthcare landscape.

If you're looking for information or assistance related to this project, could you please provide more specific details or context about what you need? Are you looking for advice on software development, project management, marketing, or something else related to Hospido or a similar healthcare app? The more information you provide, the better I can assist you.

Evaluate the importance of a well- defined Software Requirement Specification (SRS) in the software development lifecycle and its impact on project success.

- ➔ A well-defined Software Requirement Specification (SRS) is of utmost importance in the software development lifecycle as it serves as a fundamental blueprint that outlines the precise functionalities, features, and constraints of the intended software product. The SRS document provides a clear and unambiguous understanding of what needs to be developed, helping to establish a common understanding among stakeholders, including developers, testers, and clients. With a robust SRS in place, the development team can effectively plan, design, and implement the software, reducing the risk of misunderstandings, miscommunications, and scope creep. Consequently, a well-crafted SRS significantly contributes to project success by minimising rework, enhancing the efficiency of development efforts, ensuring alignment with client expectations, and ultimately delivering a high-quality software product on time and within budget.

Analyse a given SRS document to identify any ambiguities or inconsistencies and propose improvements to enhance its clarity and completeness.

- ➔ To enhance the clarity and completeness of the given SRS document, thorough analysis and improvements are essential. Firstly, focus on using clear and unambiguous language throughout the document to eliminate any room for misinterpretation. Revisit each requirement and address any inconsistencies or conflicts among them, ensuring that they align seamlessly. Additionally, verify the document's completeness by cross-referencing with stakeholders and double-checking that no crucial requirements have been overlooked. Establish requirements traceability by uniquely identifying and linking each requirement to relevant sections and artifacts. Review the provided use case scenarios to ensure they encompass the full spectrum of user interactions and system responses. If necessary, propose new use cases to further enhance the SRS document's comprehensiveness. Validate the feasibility and realism of the stated requirements, making necessary modifications to avoid impractical or overly complex elements. Lastly, fix any cross-referencing errors and ensure accurate references to other sections or documents. By following these steps, the SRS document's quality will be significantly improved, reducing misunderstandings and contributing to a more successful software development process.

Compare and contrast different techniques for requirement elicitation, such as interviews, surveys, and use case modelling, and determine their effectiveness in gathering user needs.

- ➔ Requirement elicitation is a crucial phase in software development, involving various techniques to gather user needs effectively. Interviews, surveys, and use case modelling are three common techniques used for this purpose. Interviews involve direct one-on-one interactions with stakeholders, allowing for in-depth discussions and clarifications. Surveys, on the other hand, enable the collection of a large amount of data from a broader audience, although they may lack the depth of insights provided by interviews. Use case modelling focuses on understanding user interactions with the system through scenario-based descriptions. Each technique has its strengths and weaknesses. Interviews excel at providing rich and contextual information but can be time-consuming and limited to a smaller sample size. Surveys are efficient for gathering data from a larger user base but may lack in-depth explanations. Use case modelling facilitates visualising system functionality but might not capture all user nuances. The effectiveness of each technique depends on the specific project's requirements, user base, and resources available, and a combination of these techniques is often employed to ensure a comprehensive understanding of user needs.